Arithematic Operations

May 12, 2025

1 Integers

[7]:
```
print('Addition: ', 1 + 2)
```

```
Addition:  3
```

[9]:
```
print('Subtraction: ', 2 - 1)
```

```
Subtraction:  1
```

[11]:
```
print('Multiplicattion:', 2*3)
```

```
Multiplicattion: 6
```

[13]:
```
print('Division:', 4/2) # Division in python gives floating number
```

```
Division: 2.0
```

[15]:
```
print('Division:', 4//2)# gives without the floating number or without the␣
↪remaining
```

```
Division: 2
```

[19]:
```
print('Exponential:',3**3)#Power
```

```
Exponential: 27
```

2 Floating Numbers

[22]:
```
print('Floating Number,PI', 3.14)
```

```
Floating Number,PI 3.14
```

[24]:
```
print('Floating Number, gravity', 9.81)
```

```
Floating Number, gravity 9.81
```

3 Complex Numbers

[31]:
```
print('Complex number:', 1+1j)
```

```
Complex number: (1+1j)
```

[35]:
```
print('Multiplying Complex Number:', (1+1j)*(1-1j))
```

```
Multiplying Complex Number: (2+0j)
```

4 Declaring the variable at the top first

[ ]:
```
a=3 # a is Variable and 3 is an integer data type
b=2 # b is Variable and 2 is an integer data type
```

4.0.1 # Arithmetic operations and assigning the result to a variable

[93]:
```
total = a + b
diff = a - b
product = a * b
division = a / b
remainder = a % b
floor_division = a // b
exponential = a ** b
```

[99]:
```
print('a + b = ', total)
print('a - b = ', diff)
print('a * b = ', product)
print('a / b = ', division)
print('a % b = ', remainder)
print('a // b = ', floor_division)
print('a ** b = ', exponential)
```

```
a + b =  5
a - b =  1
a * b =  6
a / b =  1.5
a % b =  1
a // b =  1
a ** b =  9
```

5 Declaring values and organizing them together

[102]:
```
num_one = 3
num_two = 4
```

6 Arithmetic operations

[113]:
```
total = num_one + num_two
diff = num_two - num_one
product = num_one * num_two
div = num_two / num_two
remainder = num_two % num_one
```

7 Printing values with label

[119]:
```
print('Total:', total)
print('Difference:', diff)
print('Product:', product)
print('Division:', div)
print('Remainder:', remainder)
```

```
Total: 7
Difference: 1
Product: 12
Division: 1.0
Remainder: 1
```

8 Calculating area of a circle

[123]:
```
radius = 10                                    # radius of a circle
area_of_circle = 3.14 * radius ** 2            # two * sign means exponent or power
```

[125]:
```
print('Area of a circle:', area_of_circle)
```

```
Area of a circle: 314.0
```

9 Calculating area of a rectangle

[138]:
```
length=10
breadth=20
area_of_rectangle=length*breadth
```

[140]:
```
print('Area of Rectangle:', area_of_rectangle)
```

```
Area of Rectangle: 200
```

10 Calculating a weight of an object

[145]:
```
mass = 75
gravity = 9.81
weight = mass * gravity
print('Weight:',weight, 'N')
```

```
Weight: 735.75 N
```

11 Operations

[150]:
```
print(3 > 2)
print(3 >= 2)
print(3 < 2)
print(2 < 3)
print(2 <= 3)
print(3 == 2)
print(3 != 2)    because 3 is not equal to 2
print(len('mango') == len('avocado'))
print(len('mango') != len('avocado')) (notequalto)
print(len('mango') < len('avocado'))
print(len('milk') != len('meat'))
print(len('milk') == len('meat'))
print(len('tomato') == len('potato'))
print(len('python') > len('dragon'))
```

```
True
True
False
True
True
False
True
False
True
True
False
True
True
False
```

12 Boolean comparison

[153]:
```
print('True == True: ', True == True)
print('True == False: ', True == False)
print('False == False:', False == False)
print('True and True: ', True and True)
print('True or False:', True or False)
```

```
True == True:  True
True == False:  False
False == False: True
True and True:  True
True or False: True
```

13 Another way comparison

[165]:
```
print('1 == 1', 1 == 1)                    # True - because the data values are␣
 ↪the same
print('1 is not 2', 1 is not 2)            # True - because 1 is not 2
print('A in Asabeneh', 'A' in 'Asabeneh') # True - A found in the string
print('B in Asabeneh', 'B' in 'Asabeneh') # False -there is no uppercase B
print('h in Asabeneh', 'h' in 'Asabeneh') # True - A found in the string
print('coding' in 'coding for all') # True - because coding for all has the␣
 ↪word coding
print('a in an:', 'a' in 'an')      # True
print('4 is 2 ** 2:', 4 is 2 ** 2)   # True
```

```
1 == 1 True
1 is not 2 True
A in Asabeneh True
B in Asabeneh False
h in Asabeneh True
True
a in an: True
4 is 2 ** 2: True
```

```
<>:2: SyntaxWarning: "is not" with a literal. Did you mean "!="?
<>:8: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:2: SyntaxWarning: "is not" with a literal. Did you mean "!="?
<>:8: SyntaxWarning: "is" with a literal. Did you mean "=="?
/tmp/ipykernel_593/2521026711.py:2: SyntaxWarning: "is not" with a literal. Did
you mean "!="?
  print('1 is not 2', 1 is not 2)           # True - because 1 is not 2
/tmp/ipykernel_593/2521026711.py:8: SyntaxWarning: "is" with a literal. Did you
mean "=="?
  print('4 is 2 ** 2:', 4 is 2 ** 2)   # True
```

[167]:
```
print(3 > 2 and 4 > 3) # True - because both statements are true
print(3 > 2 and 4 < 3) # False - because the second statement is false
print(3 < 2 and 4 < 3) # False - because both statements are false
print(3 > 2 or 4 > 3)  # True - because both statements are true
print(3 > 2 or 4 < 3)  # True - because one of the statement is true
print(3 < 2 or 4 < 3)  # False - because both statements are false
print(not 3 > 2)     # False - because 3 > 2 is true, then not True gives False
print(not True)      # False - Negation, the not operator turns true to false
print(not False)     # True
print(not not True)  # True
print(not not False) # False
```

```
True
False
False
True
True
False
False
False
True
True
False
```

[ ]: